```
Week 12: Capstone Project Part 5.3 NUS Salman
```

Week 12: Capstone Project Part 5.3 NUS Salman

```
!pip install transformers requests -q
```

```
import sqlite3
import requests
from datetime import datetime
from transformers import pipeline
```

```
def setup_database():
    conn = sqlite3.connect("events.db")
    c = conn.cursor()
    c.execute('''
        CREATE TABLE IF NOT EXISTS events (
            id INTEGER PRIMARY KEY,
            name TEXT,
            type TEXT,
            description TEXT,
            location TEXT,
            date TEXT
        )
    ''')
    events = [
        ('Summer Concert', 'outdoor', 'Live music in the park', 'Central Park', '2025-07-15'),
        ('Art Exhibition', 'indoor', 'Modern art showcase', 'City Gallery', '2025-07-15'),
        ('Food Festival', 'outdoor', 'International cuisine', 'Waterfront', '2025-07-16'),
        ('Theater Show', 'indoor', 'Classical drama', 'Grand Theater', '2025-07-16')
    ]
    c.executemany('INSERT INTO events (name, type, description, location, date) VALUES (?,?,?,?,?)', events)
    conn.commit()
    conn.close()

setup_database()
```

```
WEATHER_API_KEY = "d0f2e4100b804758bce175542252107"
location = "Singapore"
today = str(datetime.now().date())
summarizer = pipeline("summarization", model="sshleifer/distilbart-cnn-12-6")

def fetch_weather(location):
    print(f"Fetching weather data for {location} on {today}...")
    try:
        url = f"http://api.weatherapi.com/v1/current.json"
        params = {"key": WEATHER_API_KEY, "q": location, "aqi": "no"}
        response = requests.get(url, params=params)
        response.raise_for_status()
        return response.json()
    except Exception as e:
        print(f"Weather API error: {e}")
        return None

def fetch_events(date):
    print("Fetching events...")
    try:
        conn = sqlite3.connect("events.db")
        c = conn.cursor()
        c.execute("SELECT * FROM events WHERE date = ?", (date,))
        results = c.fetchall()
        conn.close()
        return results
    except Exception as e:
        print(f"Event DB error: {e}")
        return []

def generate_recommendation(weather_data, events):
    print("Generating recommendations...")
    if not weather_data:
        weather_info = "Weather data unavailable."
    else:
        temp = weather_data['current']['temp_c']
        cond = weather_data['current']['condition']['text']
        weather_info = f"Weather: {cond}, Temperature: {temp}°C."

    if not events:
        return "No events available today."

    event_descriptions = [f"{e[1]} ({e[2]}): {e[3]} at {e[4]}" for e in events]
    full_input = f"{weather_info}\nAvailable events:\n" + "\n".join(event_descriptions)

    summary = summarizer(full_input, max_length=100, min_length=40, do_sample=False)[0]['summary_text']
    return summary
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secret in your Google Colab and restart your session.
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
config.json:        1.80k/?  [00:00<00:00,  69.4kB/s]

pytorch_model.bin:  100%                                    1.22G/1.22G  [00:16<00:00,  98.9MB/s]

model.safetensors:   18%                                    220M/1.22G  [00:03<00:06,  148MB/s]

tokenizer_config.json:  100%                                  26.0/26.0  [00:00<00:00,  449B/s]

vocab.json:         899k/?  [00:00<00:00,  11.4MB/s]

merges.txt:         456k/?  [00:00<00:00,  11.0MB/s]
Device set to use cpu
```

```
dates = ["2025-07-15", "2025-07-16", "2025-07-17"]
for d in dates:
    print(f"===== Recommendations for {location} on {d} =====")
    weather = fetch_weather(location)
    events = fetch_events(d)
    print(generate_recommendation(weather, events))
    print()
```

```
===== Recommendations for Singapore on 2025-07-15 =====
Fetching weather data for Singapore on 2025-07-24...
Your max_length is set to 100, but your input_length is only 48. Since this is a summarization task, where outputs shorter than the input are typically wanted, you might consider decreasing max_length manually, e.g. summarizer('...', max_length=24)
Fetching events...
Generating recommendations...
Your max_length is set to 100, but your input_length is only 45. Since this is a summarization task, where outputs shorter than the input are typically wanted, you might consider decreasing max_length manually, e.g. summarizer('...', max_length=22)
 Summer Concert (outdoor): Live music in the park at Central Park . Modern art exhibition (indoor): Modern art showcase at City Gallery . Temperature: 28.3°C. Weather: Partly cloudy,   temperatures at Central Park are 28°C .

===== Recommendations for Singapore on 2025-07-16 =====
Fetching weather data for Singapore on 2025-07-24...
Fetching events...
Generating recommendations...
 Food Festival (outdoor): International cuisine at Waterfront . Theater Show (indoor): Classical drama at Grand Theater . Weather: Partly cloudy, Temperature: 28.3°C .

===== Recommendations for Singapore on 2025-07-17 =====
Fetching weather data for Singapore on 2025-07-24...
Fetching events...
Generating recommendations...
No events available today.
```